

In the Claims:

1. (Currently amended) A method of providing server affinities for related connection request messages in networking environments which perform workload balancing, comprising ~~steps of~~:

signaling, with ~~by~~ an executing server application, that an affinity with a selected source is to be started; and

bypassing normal workload balancing operations, ~~responsive in response~~ to the signaling, for subsequent connection request messages from the selected source while the affinity persists.

2. (Original) The method according to Claim 1, wherein the selected source is a selected client.

3. (Original) The method according to Claim 2, wherein the selected client is identified by its Internet Protocol ("IP") address.

4. (Original) The method according to Claim 2, wherein the selected client is identified by its Internet Protocol ("IP") address and port number.

5. (Original) The method according to Claim 1, wherein the selected source is a selected client subnetwork.

6. (Currently amended) The method according to Claim 1, further comprising ~~the step of~~ signaling, by the executing server application, that the started affinity with the selected source is to be ended; and wherein ~~the step of~~ bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

7. (Currently amended) The method according to Claim 1, wherein the started affinity persists for a maximum duration, after which ~~the step of~~ bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

8. (Original) The method according to Claim 7, wherein the executing server application may override the maximum duration when signaling the start of the affinity.

9. (Currently amended) The method according to Claim 7, wherein, prior to expiration of the maximum duration for which the started affinity persists, each of the subsequent connection request messages automatically extends the maximum duration of the started affinity.

10. (Currently amended) The method according to Claim 9, further comprising ~~the step of~~ extending, by the executing server application prior to expiration of the maximum duration for which the started affinity persists, the started affinity beyond the maximum duration.

11. (Currently amended) The method according to Claim 1, wherein ~~the bypassing normal workload balancing operations step~~ causes the subsequent connection request messages from the selected source to be routed to an instance of the executing server application which signaled the affinity start.

12. (Currently amended) A method of routing related connection requests in a networking environment which performs workload balancing, comprising ~~steps of~~:

storing information for enforcing one or more currently-active affinities, responsive to receiving start affinity requests for each such currently-active affinity from one or more executing server applications;

receiving incoming connection requests from client applications; and

routing each received connection request to a proper one of the executing server applications, further comprising ~~steps of~~:

selecting a particular one of the executing server applications using the stored information for enforcing affinities, when the client application sending the received connection request is identified in the stored information as having one of the one or more currently-active affinities with the particular one of the executing server applications; and

selecting the particular one of the executing server applications using workload balancing ~~otherwise~~ when the client application sending the received connection request is not identified in the stored information as having one of the one or more currently-active affinities with the particular one of the executing server applications.

13. (Original) The method according to Claim 12, wherein the client application is identified as having one of the currently-active affinities with the particular one if a destination address and destination port, as well as a source address and optionally a source port number, of the connection request being routed match the stored information.

14. (Currently amended) The method according to Claim 12, further comprising ~~the step of~~ removing stored information for enforcing selected ones of the currently-active affinities, responsive to receiving an end affinity request from selected ones of the executing server applications which stored the information.

15. (Currently amended) The method according to Claim 12, further comprising ~~the step of~~ removing stored information for enforcing selected ones of the currently-active affinities, responsive to expiration of a duration value for the selected ones.

16. (Currently amended) A system for providing server affinities for related connection request messages in networking environments which perform workload balancing, comprising:

means for signaling, with ~~by~~ an executing server application, that an affinity with a selected source is to be started; and

means for bypassing normal workload balancing operations, ~~responsive in response~~ to the signaling, for subsequent connection request messages from the selected source while the affinity persists.

17. (Original) The system according to Claim 16, further comprising means for signaling, by the executing server application, that the started affinity with the selected source is to be ended; and wherein the means for bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

18. (Original) The system according to Claim 16, wherein the started affinity persists for a maximum duration, after which the means for bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

19. (Original) The system according to Claim 18, wherein the executing server application may override the maximum duration when signaling the start of the affinity.

20. (Currently amended) The system according to Claim 18, wherein, prior to expiration of the maximum duration for which the started affinity persists, each of the subsequent connection request messages automatically extends the maximum duration of the started affinity.

21. (Currently amended) The system according to Claim 20, further comprising means for extending, by the executing server application prior to expiration of the maximum duration for which the started affinity persists, the started affinity beyond the maximum duration.

22. (Original) The system according to Claim 16, wherein the means for bypassing causes the subsequent connection request messages from the selected source to be routed to an instance of the executing server application which signaled the affinity start.

23. (Currently amended) A system for routing related connection requests in a networking environment which performs workload balancing, comprising:

means for storing information for enforcing one or more currently-active affinities, responsive to receiving start affinity requests for each such currently-active affinity from one or more executing server applications;

means for receiving incoming connection requests from client applications; and

means for routing each received connection request to a proper one of the executing server applications, further comprising:

means for selecting a particular one of the executing server applications using the stored information for enforcing affinities, when the client application sending the received connection request is identified in the stored information as having one of the one or more currently-active affinities with the particular one of the executing server applications;
and

means for selecting the particular one of the executing server applications using workload balancing ~~otherwise~~ when the client application sending the received connection request is not identified in the stored information as having one of the one or more currently-active affinities with the particular one of the executing server applications.

24. (Original) The system according to Claim 23, wherein the client application is identified as having one of the currently-active affinities with the particular one if a destination address and destination port, as well as a source address and optionally a source port number, of the connection request being routed match the stored information.

25. (Original) The system according to Claim 23, further comprising means for removing stored information for enforcing selected ones of the currently-active affinities, responsive to receiving an end affinity request from selected ones of the executing server applications which stored the information.

26. (Original) The system according to Claim 23, further comprising means for removing stored information for enforcing selected ones of the currently-active affinities, responsive to expiration of a duration value for the selected ones.

27. (Currently amended) A computer program product for providing server affinities for related connection request messages in networking environments which perform workload balancing, the computer program product embodied on one or more computer readable media and comprising:

computer readable program code means for signaling, with ~~by~~ an executing server application, that an affinity with a selected source is to be started; and

computer readable program code ~~means~~ for bypassing normal workload balancing operations, responsive in response to the signaling, for subsequent connection request messages from the selected source while the affinity persists.

28. (Currently amended) The computer program product according to Claim 27, further comprising computer readable program code ~~means~~ for signaling, by the executing server application, that the started affinity with the selected source is to be ended; and wherein the computer readable program code ~~means~~ for bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

29. (Currently amended) The computer program product according to Claim 27, wherein the started affinity persists for a maximum duration, after which the computer readable program code ~~means~~ for bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

30. (Original) The computer program product according to Claim 29, wherein the executing server application may override the maximum duration when signaling the start of the affinity.

31. (Currently amended) The computer program product according to Claim 29, wherein, prior to expiration of the maximum duration for which the started affinity persists, each of the subsequent connection request messages automatically extends the maximum duration of the started affinity.

32. (Currently amended) The computer program product according to Claim 30, further comprising computer readable program code ~~means~~ for extending, by the executing server application prior to expiration of the maximum duration for which the started affinity persists, the started affinity beyond the maximum duration.

33. (Currently amended) The computer program product according to Claim 27, wherein the computer readable program code ~~means~~ for bypassing causes the subsequent

connection request messages from the selected source to be routed to an instance of the executing server application which signaled the affinity start.

34. (Currently amended) A computer program product for routing related connection requests in a networking environment which performs workload balancing, the computer program product embodied on one or more computer readable media and comprising:

computer readable program code means for storing information for enforcing one or more currently-active affinities, responsive to receiving start affinity requests for each such currently-active affinity from one or more executing server applications;

computer readable program code means for receiving incoming connection requests from client applications; and

computer readable program code means for routing each received connection request to a proper one of the executing server applications, further comprising:

computer readable program code means for selecting a particular one of the executing server applications using the stored information for enforcing affinities, when the client application sending the received connection request is identified in the stored information as having one of the one or more currently-active affinities with the particular one of the executing server applications; and

computer readable program code means for selecting the particular one of the executing server applications using workload balancing otherwise when the client application sending the received connection request is not identified in the stored information as having one of the one or more currently-active affinities with the particular one of the executing server applications.

35. (Original) The computer program product according to Claim 34, wherein the client application is identified as having one of the currently-active affinities with the particular one if a destination address and destination port, as well as a source address and optionally a source port number, of the connection request being routed match the stored information.

36. (Currently amended) The computer program product according to Claim 34, further comprising computer readable program code ~~means~~ for removing stored information for enforcing selected ones of the currently-active affinities, responsive to receiving an end affinity request from selected ones of the executing server applications which stored the information.

37. (Currently amended) The computer program product according to Claim 34, further comprising computer readable program code ~~means~~ for removing stored information for enforcing selected ones of the currently-active affinities, responsive to expiration of a duration value for the selected ones.